

REMARKS

The holding of allowable subject matter is gratefully acknowledged.

IMPROPER REFUSAL TO ENTER AMENDMENT UNDER RULE 116

On July 29, 2005, Applicant submitted an amendment under section 116, adding new claims in the expectation that the erroneous final office action would be withdrawn and prosecution re-opened. The finality of the office action of June 13, 2005 has indeed been withdrawn, the Examiner apparently having been convinced that it was erroneous. Accordingly, there was no legal basis for refusal to enter and consider the new claims presented on July 29. They should have been entered and considered. Correction of this error is respectfully requested.

In particular, Applicant respectfully submits that it would be improper for the Examiner to issue a final office action on claims 14-18 in response to the present amendment, because they should have been given a non-final examination in the office action of April 18, 2006.

Rejection under section 101

The claims have been amended to put them in a more traditional format. Applicant respectfully submits that these changes are purely formal in nature and do not affect the scope of the claims.

REMARKS

Other claim amendments

Claims 12 and 13 have been amended to correct minor usage errors: changing “a” to —an— and inserting the word —of—. These changes do not affect the scope of the claims and were not in response to any rejection.

Art rejections

The art rejections are respectfully traversed.

Since the references are several and complex, Applicant will confine his remarks to those portions of the references cited by the Examiner, except as otherwise indicated. Applicants make no representation as to the contents of other portions of the references.

Any of the Examiner’s rejections and/or points of argument that are not addressed below would appear to be moot in view of the following. Nevertheless, Applicant reserves the right to respond to those rejections and arguments and to advance additional arguments at a later date. No arguments are waived and none of the Examiner’s statements are conceded.

In the prior prosecution, Applicant has repeatedly explained to the Examiner what object-oriented (o-o) programming is and how Applicant has carefully defined what an object is in the specification. In particular, the nature of objects as software creations

REMARKS

having both data and executable code (bottom of p. 2 of the specification) in them has been explained.

Just as a word of explanation, the undersigned would like to insert her personal understanding of o-o programming. The undersigned learned programming back in the late 1970's. At that time, the craze was "structured programming." In structured programming, a program was divided up into modules called "procedures" that could call each other. The procedures were not supposed to share data, except by passing parameters, but sometimes they did, depending on how meticulous the programmer was. This could give rise to problems, especially when a number of programmers were working together and they weren't aware that some procedures were cheating. The procedures were also called sequentially.

The undersigned understands o-o programming to be a successor to that concept, where instead of "procedures," there are "objects." The "objects" are much stricter. They can only communicate using messages. They cannot reach into the data inside another object and modify it. Also, they are not called sequentially. They exist in parallel to each other – though new ones can be created. This makes objects better able to work in a parallel processing environment, without the programmer necessarily being aware of the hardware.

Given the very lengthy nature of the prosecution here and the unfortunate necessity

REMARKS

there has been for Applicant to write TWO appeal briefs, both of which have resulted in withdrawal of the respective final office actions, Applicant would have hoped that the Examiner by now had understood the point that Applicant was making. In view of all the time that has been wasted to date on this application, Applicant implores the Examiner to review the prior prosecution and understand what objects are and what object-oriented programming is.

Despite the prior history, once again, the Examiner has cited references, Klausmeier and Bubenik, neither of which is an o-o reference; and characterized items which are not objects as objects.

Claim 1

The Examiner states that the “cells” of Klausmeier are objects. Applicant respectfully disagrees. At col. 1, lines 15-18, Klausmeier defines cells as units of information, data only. They are not data objects, as they only contain data, not executable code.

The Examiner states that Klausmeier transfers data to “processing objects” at col. 2, lines 25-30. Applicant sees no processing objects here at all. There are queues, but queues alone are not what Applicant is claiming. As far as Applicant can tell, Klausmeier teaches a communications network where cells of information are transmitted amongst physical

REMARKS

destinations. In this respect, Klausmeier appears to be similar to the Zhou reference that the Examiner cited so long ago.

The Examiner appears to purport to find a path object in Klausmeier's Fig. 7. Fig. 7 is a queue controller. It does contain an array of queues. But Applicant is not finding any object here. Fig. 7 appears to show a hardware device, per the last paragraph in col. 4. An object is a creation of software. The array of queues appears to be either a data structure or a memory device. It does not appear to be an object, because it contains only data, not executable code. It appears that any number of programs might have access to the data in these data structures or memory devices – contrary to the nature of objects in o-o programming.

Various parts of columns 5, 13 and 15 are cited. These confirm that the cells and queue array are only for data storage and are not objects. Applicant has reviewed the sections and finds no indication that the queues are for process objects or that any objects are involved at all.

To correct the deficiencies of Klausmeier, the Examiner cites Bubenik. Again this is a reference that relates to sending messages through a network of hardware devices – like Zhou and Klausmeier. Cells, which are data, are sent to ports, which are hardware. Applicant is not finding anything about objects in Bubenik at all. The Examiner appears to confuse a processor with a processing object. The two are not the same. A processor is a

REMARKS

hardware device. A processing object is a software device, resident in at least one hardware device.

Similar arguments may be advanced with respect to the other independent claims.

In view of the above, Applicant respectfully submits that the Examiner has failed to make a *prima facie* case of obviousness here.

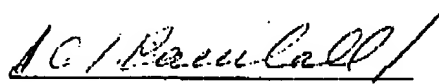
Please charge any fees other than the issue fee to deposit account 14-1270.

Please credit any overpayments to the same account.

REMARKS

Applicant respectfully submits that he has addressed each issue raised by the Examiner — except for any that were skipped as moot — and that the application is accordingly in condition for allowance. Allowance is therefore respectfully requested.

Respectfully submitted,



By _____

Anne E. Barschall
Reg. No. 31,089
(914) 332-1019
fax 914-332-7719
July 14, 2006

In-house attorney at assignee:
Michael Belk
Reg. No. 33,357
914-333-9643